جَامعة المَنارة
MANARA UNIVERSITY

# وسائط متعددة
# المحاضرة الثامنة

د. أحمد أحمد
أ.د. فادي غصنه

# أنواع الصور

**1- الصور (1-bit image) أو الصور الثنائية (binary image) أو أحادية اللون (Monochrome):**

- يتم تخزين كل بكسل على هيئة بت واحد (٠ أو ١).
- مثال: صور الفاكس.
- مثال: تتطلب الصورة أحادية اللون بدقة ٦٤٠×٤٨٠
مساحة تخزين تبلغ ٣٨,٤ كيلوبايت:

$$\frac{480 * 640}{8} = 38.4\ KB$$

Standard Lena image

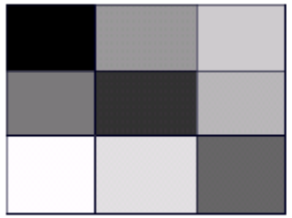
Monochrome 1-bit Lena image

# أنواع الصور

2- الصور بمستوى رمادي ٨ بت (Gray-Level Images) :

- يتم تمثيل كل بكسل بواسطة بايت واحد (كل بكسل له قيمة رمادية بين ٠ و ٢٥٥).
- يمكن اعتبار الصورة مصفوفة ثنائية الأبعاد من قيم البكسل، نسميها خريطة نقطية (bitmap).
- تتطلب الصورة ذات التدرج الرمادي مقاس ٦٤٠×٤٨٠ مساحة تخزين تبلغ ٣٠٠ كيلوبايت.

8-Bit Gray-Level Image VS 1-Bit Image

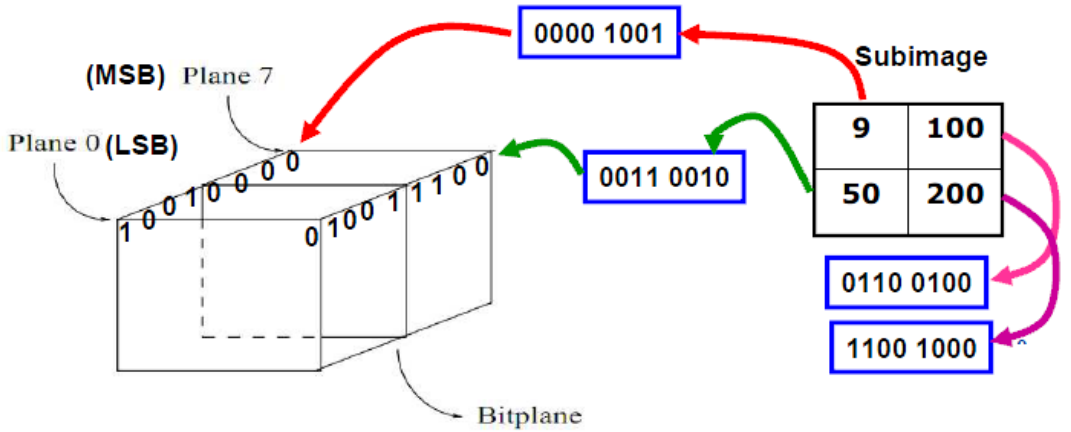


$$I = \begin{bmatrix} 0 & 150 & 200 \\ 120 & 50 & 180 \\ 250 & 220 & 100 \end{bmatrix}$$

# أنواع الصور

2- الصور بمستوى رمادي ٨ بت (Gray-Level Images) :

- يمكننا اعتبار الصورة الرمادية كمجموعة من عدة مستويات ذات البت الواحد (1-bit image).
  - ✓ لكل مستوى بت قيمة ٠ أو ١ عند كل بكسل.
  - ✓ تشكل جميع مستويات البتات معًا بايتًا واحدًا يخزن قيمًا بين ٠ و٢٥٥.




Bitplanes for 8-bit grayscale image

# أنواع الصور

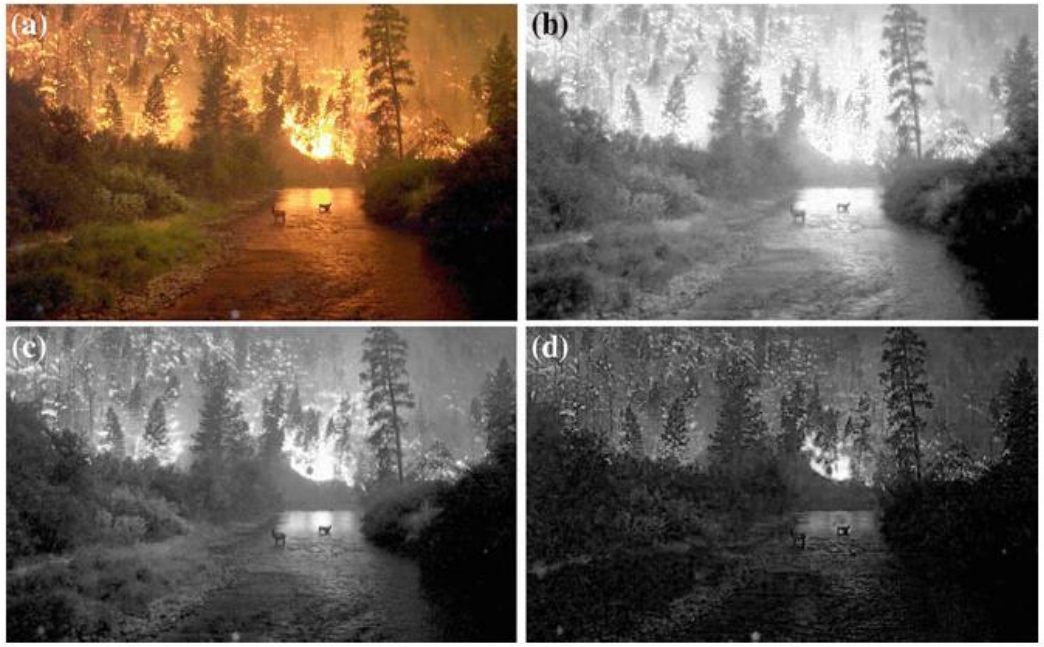
3- الصور الملونة ٢٤ بت (RGB Images) :

- يتم تمثيل كل بكسل بثلاثة بايتات.
- نظرًا لأن كل قيمة لونية تقع في النطاق من ٠ إلى ٢٥٥، فإن هذا النوع يدعم ٢٥٦×٢٥٦×٢٥٦، أو ما مجموعه ١٦,٧٧٧,٢١٦ لونًا مجمعًا ممكنًا.
- تتطلب صورة ملونة بدقة ٦٤٠ × ٤٨٠ و٢٤ بت للتخزين ٩٢١.٦ كيلوبايت.
- ملاحظة: الصورمتعددة الأطياف / فائقة الطيف (Multispectral / hyperspectral) تكون مكونة من أكثرمن ثلاثة ألوان؛ مثال : صورالأقمار الصناعية.

$$R=\begin{bmatrix}255 & 240 & 240\\ 255 & 0 & 80\\ 255 & 0 & 0\end{bmatrix} \quad G=\begin{bmatrix}0 & 160 & 80\\ 255 & 255 & 160\\ 0 & 255 & 0\end{bmatrix} \quad B=\begin{bmatrix}0 & 80 & 160\\ 0 & 0 & 240\\ 255 & 255 & 255\end{bmatrix}$$

# أنواع الصور

## 3- الصور الملونة ٢٤ بت (RGB Images) :




High-resolution color and separate R, G, B color channel images. a example of 24-bit color image `forestfire.bmp`; (**b**, **c**, **d**) R, G, and B color channels for this image

# أنواع الصور

4- الصور الملونة ٨ بت :

- من الممكن الحصول على صور ملونة دقيقة إلى حد معقول عن طريق تحديد كمية المعلومات اللونية مع ٨ بتات فقط من المعلومات (ما يسمى بـ ٢٥٦ لونًا).
- تستخدم ملفات الصور الملونة (٨ بت) جدول البحث (lookup table) لتخزين المعلومات.
- لا تخزن الصورة اللون، بل تخزن مجموعة من البايتات فقط. كل بايت هو فهرس لجدول يحتوي على ثلاث قيم (٣بايتات) تحدد اللون المكون من ٢٤ بت لبيكسل مع فهرس جدول البحث هذا.
- الفكرة المستخدمة في الصور الملونة ذات ٨ بت هي تخزين الفهرس لكل بكسل فقط.

✓ أي: إذا كان ٢٣ بكسلًا في صورة (24 bits) يحتوي على قيم (RGB: 45,200,91)، يتم تخزين القيمة ٢٣ في مصفوفة الصورة، ويتم ربط العنصر المفهرس (٢٣) في الجدول بقيم الفهرس [٤٥، ٢٠٠، ٩١].

# أنواع الصور

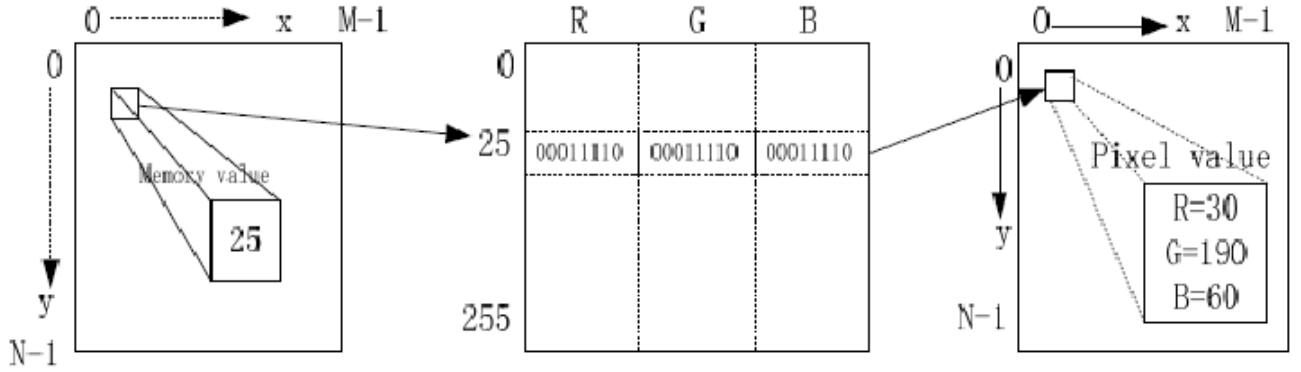
## 4- الصور الملونة ٨ بت :

**24-bit Color Image**

A 640 x 480 24-bit colour image only requires 921.6 kB of storage

**8-bit Color Image**

A 640 x 480 8-bit colour image only requires 307.2 kB of storage (without LUT)




**Value as the Index to Table**

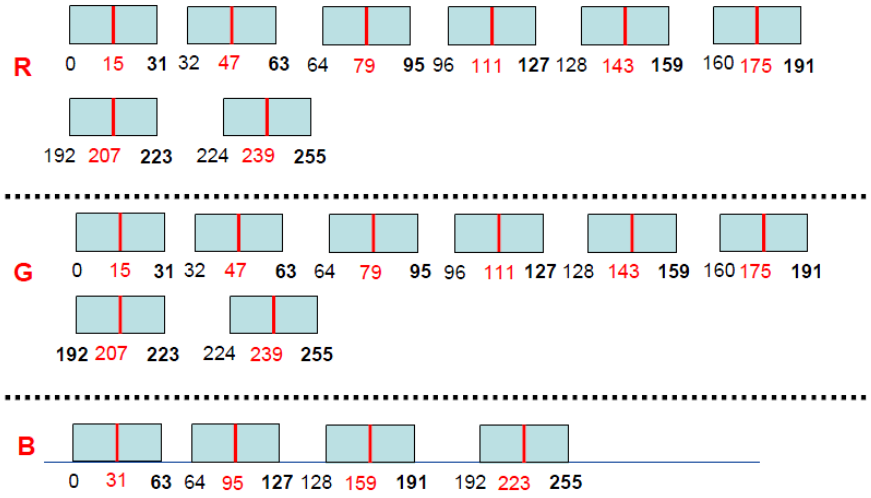
**Get the color values by Searching**

**The RGB value of the pixel**

# أنواع الصور

## 4- استخدام جدول البحث الصور الملونة ٨ بت :

- قسّم الثلاثية RGB إلى شرائح متساوية في كل بُعد، المركبة R و المركبة Gالى ٨ مجالات كل منها، مع تحديد مراكز المجالات، أما المركبة B الى ٤ مجالات مع تحديد مراكز تلك المجالات أيضا، نظرًا لأن البشر أكثر حساسية لـ Rو Gمن B.




**24-bit colour image**

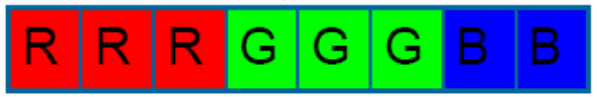
| R | | G | | B | |
|---|---|---|---|---|---|
| 45 | 100 | 200 | 190 | 70 | 100 |
| 239 | 160 | 15 | 240 | 239 | 160 |

**8-bit colour image**

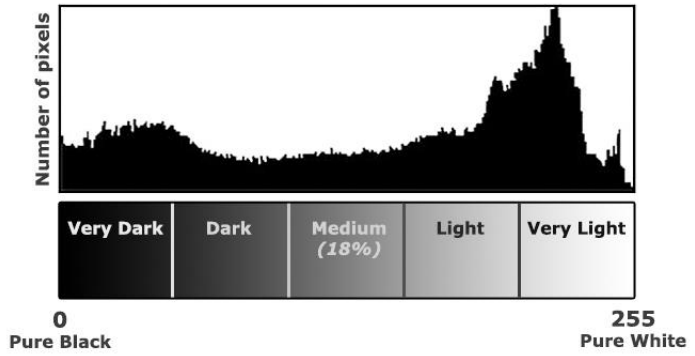
| | |
|---|---|
| 0 | 1 |
| 2 | 3 |

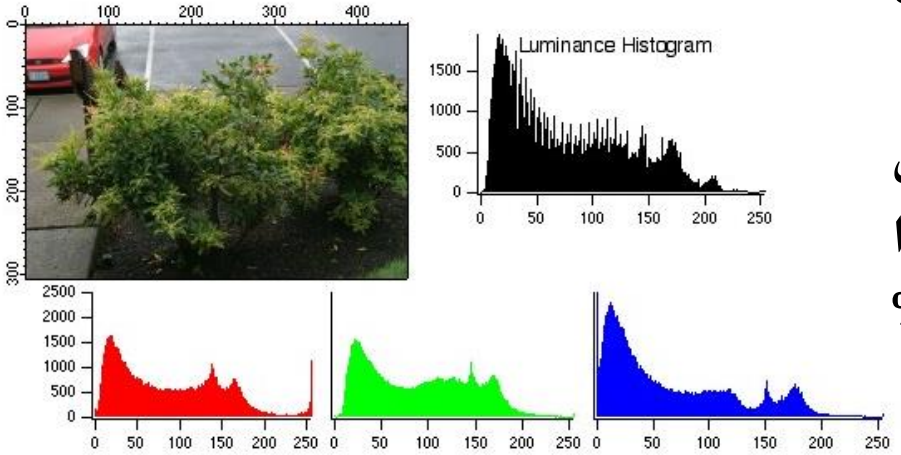
| | R | G | B |
|---|---|---|---|
| 0 | 47 | 207 | 95 |
| 1 | 111 | 175 | 95 |
| 2 | 239 | 15 | 223 |
| 3 | 175 | 239 | 159 |

- Example: 8-Bit Color Image
  - Humans are more sensitive to R and G than to B
  - So R=3, G=3 and B=2



# The Histogram الهيستوجرام




- الهيستوجرام: هو تمثيل رسومي للبكسلات المعروضة في الصورة (عدد البكسلات التي تحمل نفس القيمة)، حيث يمثل الهيستوجرام سطوع (luminosity) كل بكسل في الصورة.
- يمكن تمثيل الهيستوجرام لكل مركبة لونية بشكل منفرد، أو بتمثيل هيستوجرام السطوع ( luminance histogram).
- حساب هيستوجرام السطوع : يُمثل كل بكسل سطوعًا معيناً بناءً على متوسط مُرجّح للألوان الثلاثة لديه (الأحمر، الأخضر، والأزرق). ويفترض هذا الترجيح أن الأخضر يُمثل ٥٩% من السطوع المُدرَك، بينما يُمثل الأحمر ٣٠% فقط، والأزرق ١١%، تجمع القيم الثلاثة بعد الترجيح لتنتج قيمة السطوع.

# فضاءات الألوان

**فضاءات الألوان الشائعة:**

1- RGB ويستخدم بشكل أساسي في computer graphics

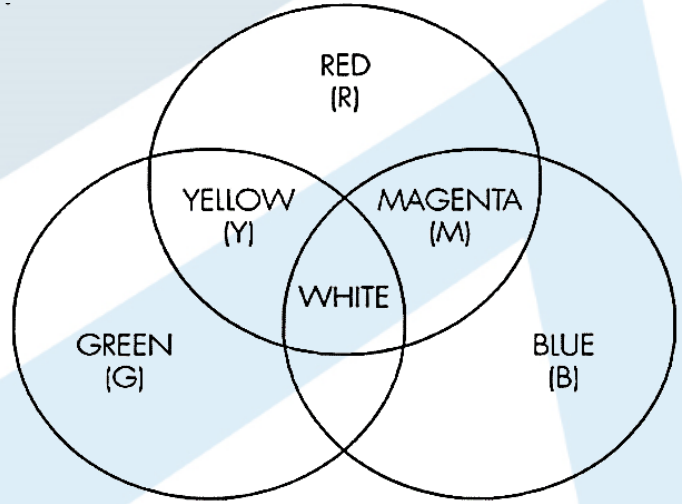
2- YIQ or YUV or YCbCr وتستخدم في Video System

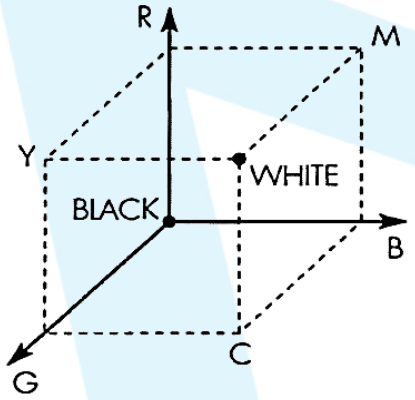
3- CMY ويستخدم في الطباعة الملونة color printing

كل فضاءات الألوان يمكن اشتقاقها من معلومات RGB التي تكون موجودة في الأجهزة مثل (scanner, cameras)

# فضاء الألوان RGB

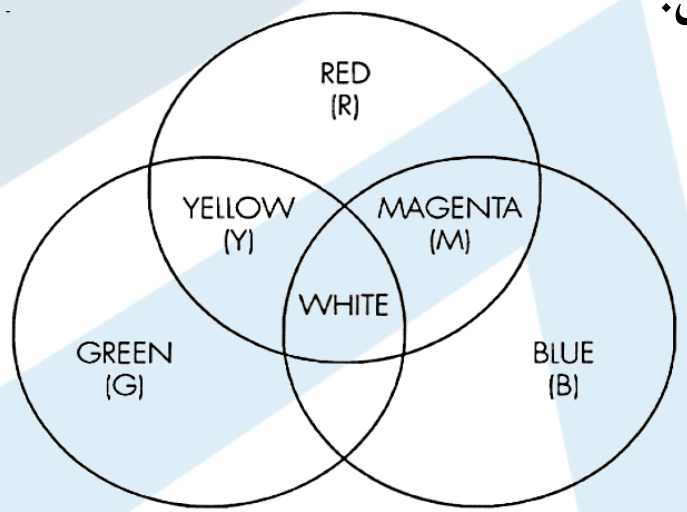
و تشكل الألوان الثلاثة (أحمر، أخضر، أزرق) أساس هذا النظام و لتشكيل الألوان المختلفة نقوم بمزج هذه الألوان الثلاثة بكميات متفاوتة حيث تمثل هذه الألوان عن طريق نظام إحداثيات ثلاثي الأبعاد لنحصل على مكعب بحيث يمثل قطر المكعب كمية متساوية من الألوان الثلاثة و التي تمثل مستويات متعددة من اللون الرمادي.

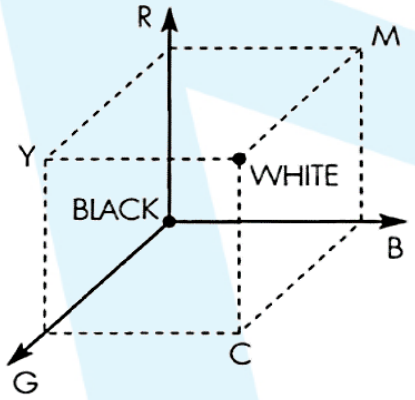


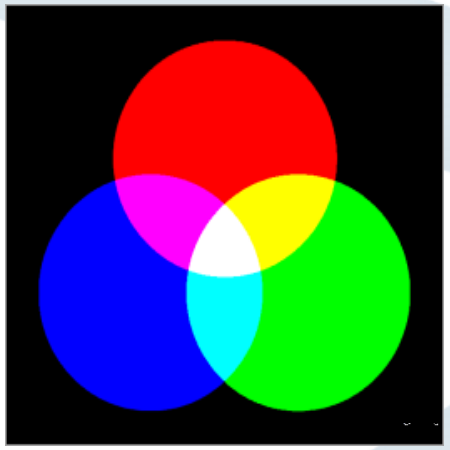
# فضاء الألوان RGB

تقنية المزج اللوني الجمعي، حيث يتم إنتاج الأسود عندما تكون جميع الألوان الأساسية الثلاثة صفراً (R, G & B=0)، وهو مفيد بشكل خاص لإنتاج صورة ملونة على سطح أسود كما هو الحال في تطبيقات العرض.



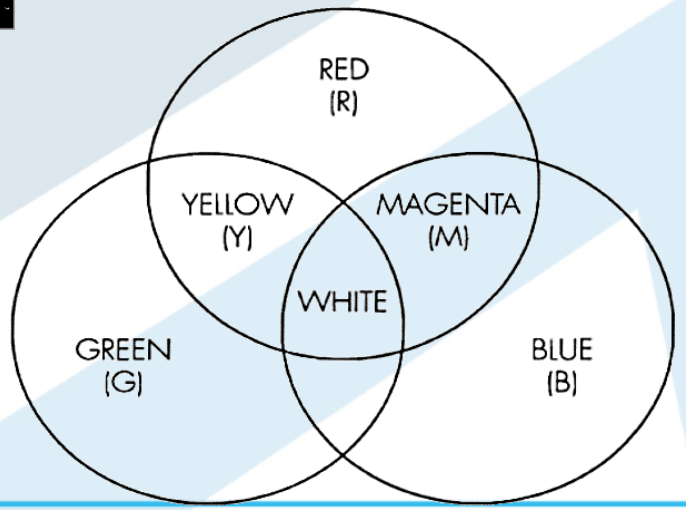



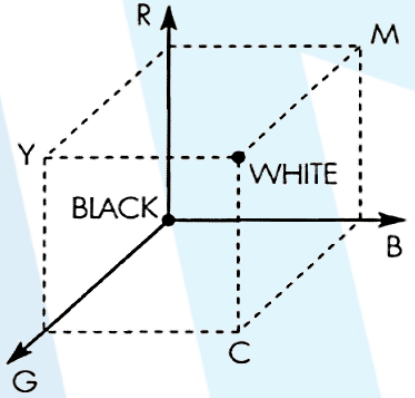
# فضاء الألوان RGB



أسباب انتشار هذا النظام:

- يبسط هذا الفضاء بنية و تصميم النظام.
- عن طريق الألوان الثلاثة الأساسية يمكن الحصول على أي لون نرغب .
- وجود هذا الفضاء منذ سنوات عديدة مما يمكننا من الإستفادة من العدد الضخم من الروتينات و الأنظمة المصممة اعتمادا عليه.



# فضاء الألوان RGB

ولكن من ناحية أخرى RGB لا يستخدم مع real-world images لأن:

Buffer Frame يملك العمق اللوني نفسه لكل بيكسل و بالتالي يقوم بإظهار الدقة نفسها لكل العناصر فمثلا لتغيير كثافة أو لون بيكسل معين من الصورة يجب بداية أن نقرأ قيم الألوان الثلاثة من Frame Buffer ثم نقوم بحساب الكثافة أو اللون و من ثم نقوم بإنجاز التعديلات المطلوبة و بعدها نقوم بحساب القيم الجديدة لل RGB ومن ثم إعادة كتابتها في Buffer Frame.

إذا استطاع النظام النفاذ بشكل مباشر إلى الصورة المخزنة وفق بنية الكثافة واللون فإن خطوات المعالجة ستكون أسهل و لهذه الأسباب فإن الفيديو القياسي يستعمل فضاءات أخرى من الألوان مثل YUV or YIQ or YCbCr وهي تتشابه مع بعضها إلا أنه يوجد فروقات.

# فضاء الألوان YUV

يستخدم من قبل:

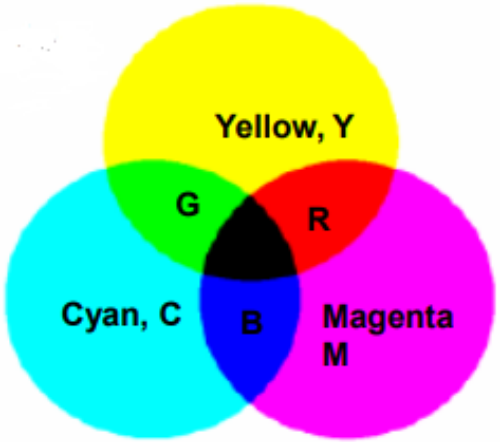
- PAL (Phase Alternation Line)
- NTSC ( National Television System Committee)
- SECAM (Sequential Color with Memory)

ويشكل هذا النظام الألوان القياسية للفيديو:

للحصول على الأبيض و الأسود نستخدم فقط معلومات Y وعند إضافة معلومات الألوان V&U نحصل على الصورة الملونة. ومن ناحية أخرى فإن هذا يتعلق بالمستقبل (القارئ) فإذا كان المستقبل يتعامل مع الأبيض والأسود فقط فإن هذا المستقبل يقوم بفك شيفرة Y وبالتالي الحصول على صورة أبيض وأسود أما بالنسبة لل Receiver الملون فهو يقوم بفك شيفرة المعلومات الأخرى وبالتالي الحصول على صورة ملونة.

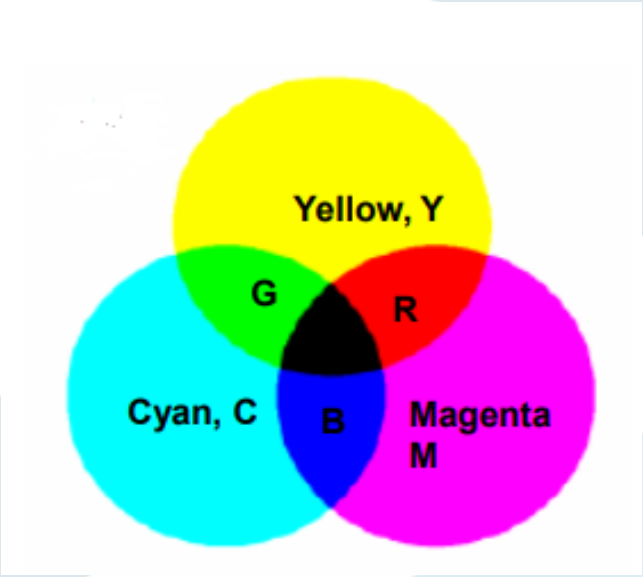
# فضاء الألوان CMYK

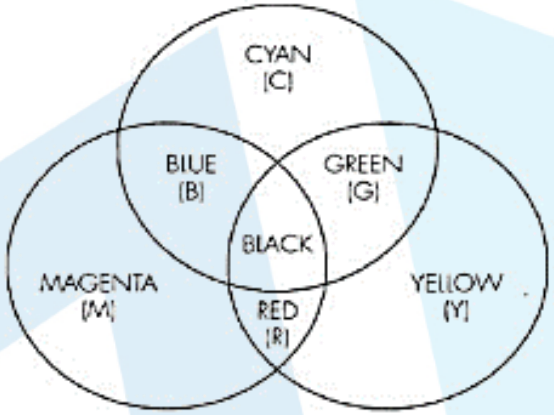
هو نموذج ألوان مطروح يستخدم في الطباعة الملونة، تستخدم معالجة الألوان هذه أربعة ألوان سماوي، وأرجواني، وأصفر، وأسود (cyan, magenta, yellow, black). يقلل الحبر المستخدم من الضوء على الورق الخلفية وهذا هو سبب تسمية هذا النموذج بالطرح لأنه يطرح يزيل السطوع من الأبيض.

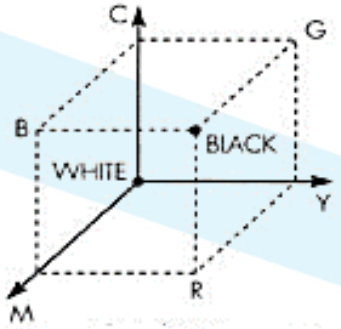


# فضاء الألوان CMYK

تقنية المزج اللوني الطرحي، ينتج اللون الأبيض عندما تكون قيم الألوان الفرعية، مساوية للصفر (C,M & Y=0)، وتستخدم من أجل الصوربخلفية بيضاء.



# بنية معطيات الصورة Graphic / Image Data Type

تتطلب عمليات عرض الصورة وحفظها وإجراء التعديلات عليها أن نكون قادرين على تخزينها بصورة متماسكة ذات تنسيق معين (هيكلية خاصة) حيث تمكننا البنى من توحيد المعايير المستخدمة في ذلك واستخدام معطيات الصورة في أكثر من تطبيق واحد .

يتم تعريف بنية Format الصورة بواسطة وسيطين:

١-الدقة المكانية Spatial resolution وتتضح في البكسلات.

٢- تشفير الألوان (أي ترميزها) Color Coding يقاس بعدد البتات لكل بكسل.

تعتمد قيمة كل من الوسيطين السابقين على ال Hard ware المستخدمين لإدخال وإخراج الصورة.

# أولاً: Graphics Interchange Format GIF

إحدى الطرق الأوسع انتشاراً ويستخدم للصور قليلة الوضوح للألوان كما في drawing & graphic وتعتبر من البنى الجيدة فهي تدعم الألوان على bit ٢٤ ولكنها محددة للوحة الألوان المكونة من ٢٥٦ لون فقط ويمكن أن يصل حجم الصورة فيها مقدراً بالبكسل إلى (64k * 64k) ويتم استخدام هذا النموذج في الحواسيب الشخصية ومحطات Workstation Unix.

يوجد نمطين لهذا النموذج :

١. GIF 87a
٢. GIF 89a

# GIF 87a

النسخة القياسية ويتم فيها تنظيم بنية الملونات كما يلي:

- GIF signature
- Screen descriptor
- Global color map
- Image descriptor
- Local color map
- Image Data
- GIF Terminator

# GIF 87a

١. **Signature**: يحدد النمط ويتكون من المحارف الستة التالية : GIF 87a.

٢. راسم الشاشة **Screen descriptor**: وهو يصف كل الوسطاء التابعة لكل صور GIF الموجودة في الملف

حيث يحدد ما يلي:

١. الأبعاد الكلية لفضاء الصورة

٢. معلومات تحويل الألوان (خريطة الألوان) الدائمة

٣. ألوان خلفية الشاشة

٤. معلومات العمق اللوني

# GIF 87a

٣. خريطة الألوان العامة (Global Color Map): تحتوي على جدول الألوان المكون من تسلسل قيم (٣ بايت) ممثلة للألوان RGB وذلك لعرض الصورة المطلوبة وتمثل هذه الخريطة لوحة الألوان الافتراضية المستخدمة لأي صورة (في هذا النوع) .

فمثلاً لدينا صورة ملونة بألوانها المحلية local map لكن كل بكسل يستقبل يتم عرضه بتقريبه إلى اللون الأقرب له في لوحة الألوان العامة global map المتاحة.

# GIF 87a

٤. **Image Descriptor** : ويحدد :

١. مكان الصورة التالية في فضاء راسم الشاشة

٢. الإشارات التي توضح خريطة الألوان المحلية

٣. تحديد تسلسل عرض البكسلات

# GIF 87a

٥. خريطة الألوان المحلية local color map: تحتوي جدول الألوان الخاصة بالصورة و الذي هو تسلسل من البايتات الممثلة للألوان الثلاثة R,G,B.

٦. معطيات الصورة: و يتكون من قيم الدليل raster image للصور المضغوطة حيث تتكون معطيات الصورة من :

١- Graphic cantor information معلومات تحكم من أجل الإحياء (animated GIFs) مثل الوقت الذي يوفق بين الصور المتعاقبة في ال animation .

٢- Animated GIFs: تعني تعريف عدة صور موجودة في ملف GIF ثم عرض هذه الصور بشكل تسلسلي.

# GIF 89a

تدعم animation لكل كتلة من المعطيات وتتميز ب:

١- توسيع التحكم بال graphics.

٢- تزود تحكم بسيط لكل زمن تأخير.

٣- تزود أدلة الشفافية.

# ثانياً: Joint Photographic Expert Group (JPEG)

تعتبر هذه البنية الأكثر شيوعاً لعرض صور بالألوان الحقيقة وتكون جودتها عالية وهي صور كاملة الألوان(true – color) ولكنها لاتشبه GIF التي حدها الأعظمي ٢٥٦ وهي تستغل محدودية الرؤية البشرية وبالتالي فإن ضياع بعض المعلومات أثناء الضغط لن يؤثرعلى المستويات المطلوبة من الجودة .

يعتبر JPEG في المعلوماتية طريقة معيارية شائعة لضغط الصور الرقمية مع خسارة في القيمة المعلوماتية للصورة ويسمى الشكل الذي يوظف هذا الضغط JPEG كذلك .

أشهر الامتدادات المستخدمة لهذا الشكل هي jpeg, jfif , jpg , JPG, JPE لكن jpg يبقى أكثرها استخداما في جميع المنصات.

# ثانياً: Joint Photographic Expert Group (JPEG)

وهذا الشكل JPEG/JFIF أكثر الأشكال استخداماً لحفظ ونقل الصور الشمسية على شبكة الويب إذ أنه مفضلة على الشكل GIF الذي لا يسمح إلا ب ٢٥٦ لون مختلف، وهذا غير كاف للصور الشمسية وPNG الذي ينتج ملفات كبيرة مقارنة ب JPEG/JFIF ويعود هذا إلى ضغطها العالي الذي يضغط البيانات بنسبة ٢٠ مرة تقريبا فمثلا إذا كانت تحتاج صورة ٢٠٠ بت فإن استعمال خوارزمية الضغط هذه يمكن تقليصها إلى ١٠ بتات.

# ثانياً: Joint Photographic Expert Group (JPEG)

خوارزمية JPEG للضغط متناظرة أي أن الخطوات اللازمة للتشفير هو نفس الخطوات اللازمة لفك التشفير

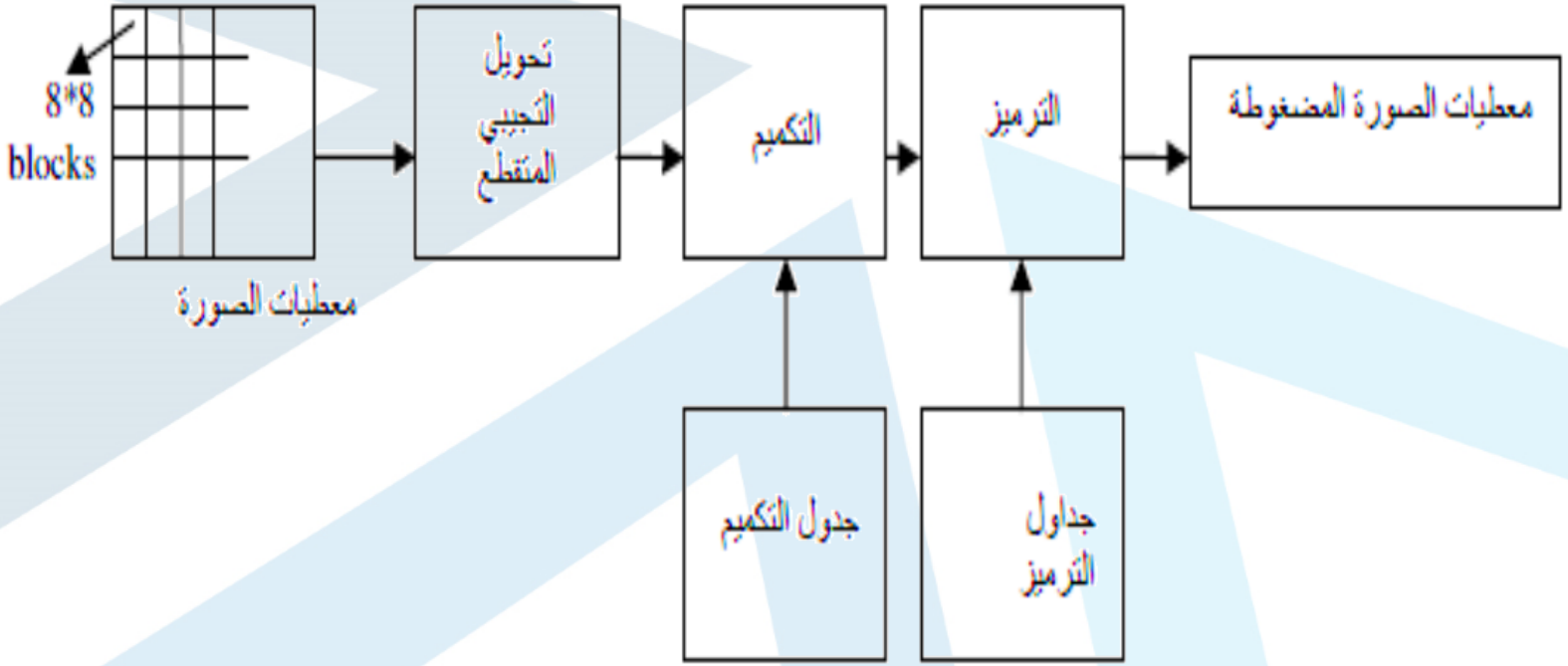
وفي ما يلي شرح لكيفية عمل الخوارزمية في أبسط صيغتها و المسماة تشفير خط الأساس المتتالي Baseline

Sequential Encoding والقائمة على الخطوات التالية :

١. تجهيز الصورة.
٢. تطبيق التحويل التجيبي المتقطع DCT.
٣. التكميم.
٤. تشكيل الترتيب ZigZag.
٥. الترميز.
٦. بناء الاطار.

# ثانياً: Joint Photographic Expert Group (JPEG)

خطوات الضغط:



# ثانياً: Joint Photographic Expert Group (JPEG)

١. تجهيز الصورة : كل صورة تمثل بمصفوفة NxM pixels. تكون عملية تجهيز الصورة بتقسيم المصفوفة الى مصفوفات جزئية يعبرعنها بالبلوك block كل منها مكون من 8x٨

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

# ثانياً: Joint Photographic Expert Group (JPEG)

٢. التحويل التجيبي المتقطع Discrete Cosine Transform (DCT):

يتطلب أولا ازاحة مستويات الرمادي ضمن المجال [-١٢٨.١٢٧]هذه العملية لا تؤمن أي ضغط للصورة، لكنها ترتيب معلومات الصورة بشكل أكثر ملائمة للضغط. يتم تطبيق تحويل DCT على كل بلوك وفق العلاقة:

$$F(u,v)=\frac{C(u)C(v)}{4}\sum_{i=0}^{7}\sum_{j=0}^{7}\cos\frac{(2i+1)u\pi}{16}\cos\frac{(2j+1)v\pi}{16}f(i,j)$$

# ثانياً: Joint Photographic Expert Group (JPEG)

حيث أن:

f : المصفوفة الجزئية , و التي أبعادها 8*8 .

F : المصفوفة الناتجة عن تطبيق التحويل DCT على المصفوفة f .

f (i ,j) : هو العنصر الموجود في السطر i و العمود j .

F(u,v) : هو العنصر الموجود في السطر u و العمود v .

I, j, u, v = 0,1,. . .7

C(u) , C(v) : ثوابت تأخذ القيم الآتية:

$C(u), C(v) = 1/\sqrt{2}$ For u, v =0 ;

$C(u), C(v) = 1$ For u, v≠ 0 ;

# ثانياً: Joint Photographic Expert Group (JPEG)

٣. التكميم: إن الهدف من التكميم هو تحقيق ضغط إضافي، حيث أن التكميم بشكل نموذجي ينتج عنه عدد كبير من العناصر التي يكون لها قيم صفرية، وهذا يؤمن معدل ضغط عالي.

يتم الحصول على القيم المكممة عن طريق ايجاد اقرب عدد صحيح لناتج كل عدد من المصفوفة F على القيمة المقابلة من مصفوفة التكميم Q وفق العلاقة:

$$F^{Q}(u,v) = IntegerRound\left(\frac{F(u,v)}{Q(u,v)}\right)$$

# ثانياً: Joint Photographic Expert Group (JPEG)

يمكن أن يكون الضغط عالي أو منخفض وذلك تبعا لجودة التكميم المستخدم كما في المثال التالي:

جدول التكميم لضغط منخفض

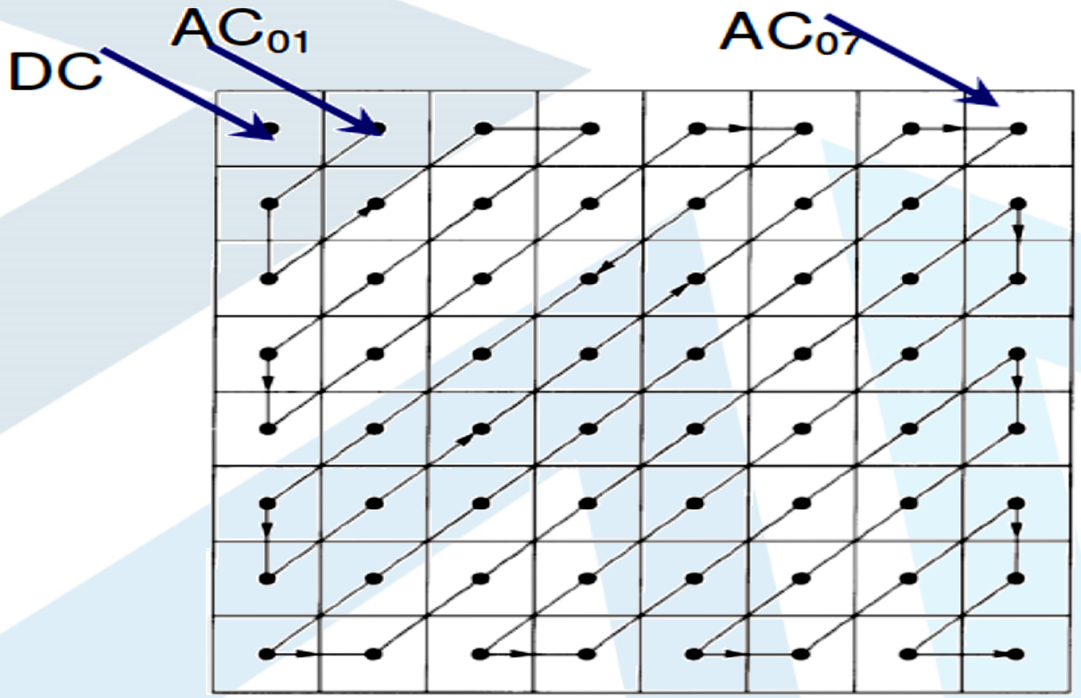
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 4 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 4 |
| 1 | 1 | 1 | 1 | 2 | 2 | 2 | 4 |
| 1 | 1 | 1 | 1 | 2 | 2 | 4 | 8 |
| 1 | 1 | 2 | 2 | 2 | 2 | 4 | 8 |
| 2 | 2 | 2 | 2 | 2 | 4 | 8 | 8 |
| 2 | 2 | 2 | 4 | 4 | 8 | 8 | 16 |
| 4 | 4 | 4 | 4 | 8 | 8 | 16 | 16 |

جدول التكميم لضغط عالي

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 |
| 2 | 4 | 4 | 8 | 16 | 32 | 64 | 128 |
| 4 | 4 | 8 | 16 | 32 | 64 | 128 | 128 |
| 8 | 8 | 16 | 32 | 64 | 128 | 128 | 256 |
| 16 | 16 | 32 | 64 | 128 | 128 | 256 | 256 |
| 32 | 32 | 64 | 128 | 128 | 256 | 256 | 256 |
| 64 | 64 | 128 | 128 | 256 | 256 | 256 | 256 |
| 128 | 128 | 128 | 256 | 256 | 256 | 256 | 256 |

# ثانياً: Joint Photographic Expert Group (JPEG)

٤. تحويل الصورة ثنائية البعد إلى شعاع وحيد البعد عن طريق قراءة كل مصفوفة جزئية بطريقة ZigZag:



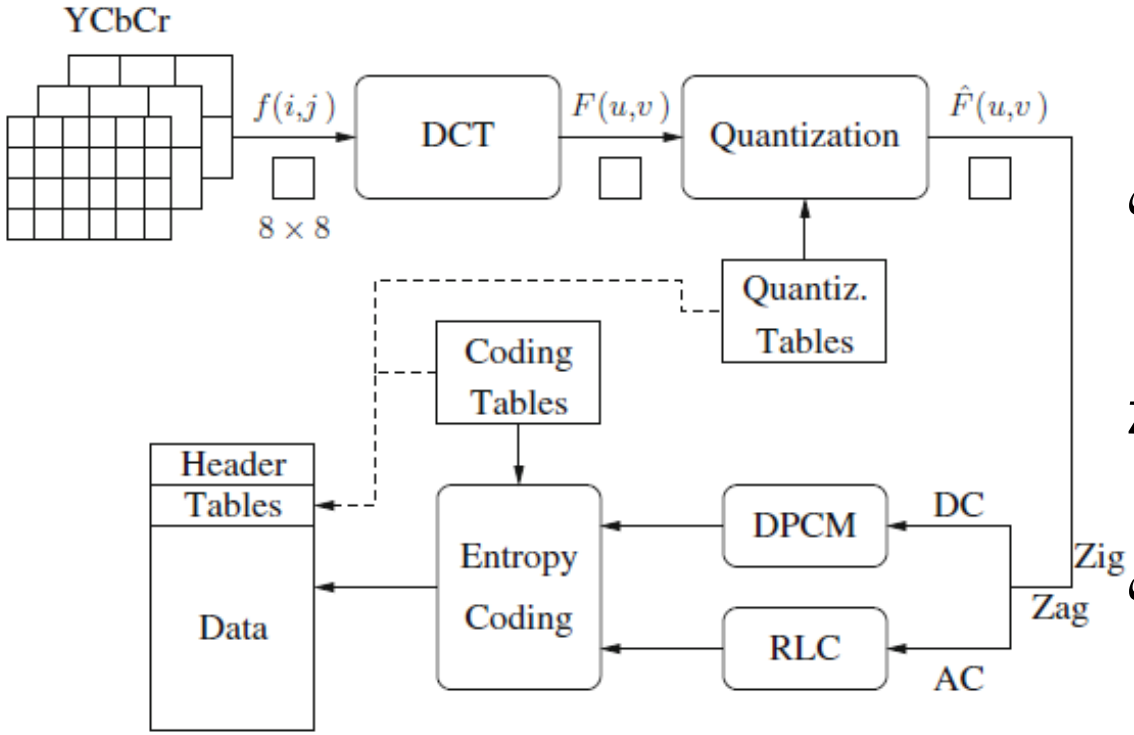
# ثانياً: Joint Photographic Expert Group (JPEG)

٥-الترميز وبناء الإطار :يتحول كل بلوك إلى مصفوفه سطرية مؤلفة من ٦٤ عنصر، يمثل العنصر الأول القيم المستمرة (DC coefficients) وباقي العناصر القيم المتناوبة (AC coefficients).

ترمز القيم المستمرة وفق ترميز هوفمان أما القيم المتناوبة ترمز وفق (Run-Length Coding(RLE.

# مثال ١ عن طريقة عمل الترميز JPEG

- كما رأينا سابقاً، فإن المخطط الصندوقي لعملية الترميز JPEG هو التالي:




- بداية يتم تمثيل الصورة في النظام اللوني YCbCr .
- يتم تقسيم الصورة الى بلوكات ٨*٨.
- يتم نقل الصورة من المجال الزمني الى المجال الترددي باستخدام التحويل DCT.
- يتم تطبيق عملية التكميم باستخدام جداول التكميم.
- يتم فرد بيانات المصفوفة الناتجة باستخدم التقنية Zig-Zag.
- يتم تطبيق المرمز DPCM على المركبة المستمرة، بينما يطبق المرمز RLC (RLE) على المركبات المتناوبة.
- يتم بعدها تطبيق الـ Entropy Coding.

# مثال ١ عن طريقة عمل الترميز JPEG

- ملاحظة١: النظام اللوني YCbCr، له مركبتين، الأولى هي السطوع Y (Luminance)، أما الثانية فهي المركبة اللونية CbCr، لكل منهما مصفوفة تكميم معيارية QM (Quantization Matrix) مستقلة.
- ملاحظة٢: يتم حساب $F(u,v)$ و هي ناتج التحويل DCT (ترددي) من خلال: $F(u,v) = \mathbf{T}.f(i,j).\mathbf{T}^T$، حيث $f(i,j)$ هي بلوك ٨*٨ في المجال الزمني، $\mathbf{T}$ مصفوفة معيارية خاصة بـ JPEG، $\mathbf{T}^T$ هو منقول المصفوفة.
- ملاحظة٣: تهدف خطوة التكميم في JPEG إلى تقليل العدد الإجمالي لبتات الصورة عن طريق التقسيم على مصفوفة التكميم و من ثم تقريب القيمة (Rounding)، $Q(u,v)$ هي مصفوفة التكميم. $\hat{F}(u,v) = \text{round}\left(\frac{F(u,v)}{Q(u,v)}\right)$
- ملاحظة٤: تحدد نسبة الضغط ( compression ratio) بمعامل الجودة QF( quality factor) والذي يأخذ القيم بين ١ و ١٠٠ حيث تمثل ١٠٠ أعلى نسبة ضغط. القيمة المعيارية لـ QF هي ٥٠ و تتوافق QM المعيارية.
- ملاحظة٥: اذا كان المطلوب الحصول عل صورة مضغوطة بدقة "QF أكبر من QF=50 فيجب حساب مصفوفة التكميم الجديدة "QM كما يلي: "QM= QM * (100- QF")/50 ، و في حال كان المطلوب صورة مضغوطة بدقة "QF أصغر من QF=50 عندئذ : "QM= QM * 50/QF.

An 8 × 8 block from the Y image of 'Lena'

يتم عرض صور السطوع (Y) فقط

المجال الزمني

المجال الترددي

بعد عملية التكميم

بعد عملية التكميم العكسي في عملية فك الضغط

بعد عملية DCT العكسي, و الرجوع الى المجال الزمني في عملية فك الضغط

الخطأ الناتج عن عملية الضغط و فك الضغط

DC

AC values

$f(i,j)$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 200 | 202 | 189 | 188 | 189 | 175 | 175 | 175 |
| 200 | 203 | 198 | 188 | 189 | 182 | 178 | 175 |
| 203 | 200 | 200 | 195 | 200 | 187 | 185 | 175 |
| 200 | 200 | 200 | 200 | 197 | 187 | 187 | 187 |
| 200 | 205 | 200 | 200 | 195 | 188 | 187 | 175 |
| 200 | 200 | 200 | 200 | 200 | 190 | 187 | 175 |
| 205 | 200 | 199 | 200 | 191 | 187 | 187 | 175 |
| 210 | 200 | 200 | 200 | 188 | 185 | 187 | 186 |

$F(u,v)$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 515 | 65 | -12 | 4 | 1 | 2 | -8 | 5 |
| -16 | 3 | 2 | 0 | 0 | -11 | -2 | 3 |
| -12 | 6 | 11 | -1 | 3 | 0 | 1 | -2 |
| -8 | 3 | -4 | 2 | -2 | -3 | -5 | -2 |
| 0 | -2 | 7 | -5 | 4 | 0 | -1 | -4 |
| 0 | -3 | -1 | 0 | 4 | 1 | -1 | 0 |
| 3 | -2 | -3 | 3 | 3 | -1 | -1 | 3 |
| -2 | 5 | -2 | 4 | -2 | 2 | -3 | 0 |

$\hat{F}(u,v)$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 32 | 6 | -1 | 0 | 0 | 0 | 0 | 0 |
| -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| -1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$\tilde{F}(u,v)$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 512 | 66 | -10 | 0 | 0 | 0 | 0 | 0 |
| -12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| -14 | 0 | 16 | 0 | 0 | 0 | 0 | 0 |
| -14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$\tilde{f}(i,j)$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 199 | 196 | 191 | 186 | 182 | 178 | 177 | 176 |
| 201 | 199 | 196 | 192 | 188 | 183 | 180 | 178 |
| 203 | 203 | 202 | 200 | 195 | 189 | 183 | 180 |
| 202 | 203 | 204 | 203 | 198 | 191 | 183 | 179 |
| 200 | 201 | 202 | 201 | 196 | 189 | 182 | 177 |
| 200 | 200 | 199 | 197 | 192 | 186 | 181 | 177 |
| 204 | 202 | 199 | 195 | 190 | 186 | 183 | 181 |
| 207 | 204 | 200 | 194 | 190 | 187 | 185 | 184 |

$\epsilon(i,j) = f(i,j) - \tilde{f}(i,j)$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 6 | -2 | 2 | 7 | -3 | -2 | -1 |
| -1 | 4 | 2 | -4 | 1 | -1 | -2 | -3 |
| 0 | -3 | -2 | -5 | 5 | -2 | 2 | -5 |
| -2 | -3 | -4 | -3 | -1 | -4 | 4 | 8 |
| 0 | 4 | -2 | -1 | -1 | -1 | 5 | -2 |
| 0 | 0 | 1 | 3 | 8 | 4 | 6 | -2 |
| 1 | -2 | 0 | 5 | 1 | 1 | 4 | -6 |
| 3 | -4 | 0 | 6 | -2 | -2 | 2 | 2 |

# مثال ٢ عن طريقة عمل الترميز JPEG

- ليكن لدينا البلوك التالي ٨*٨ من صورة ما (١).
- نظرًا لأن DCT يعمل مع قيم بكسل تتراوح من -١٢٨ إلى +١٢٧، فيمكننا طرح ١٢٨ من المصفوفة الأصلية، مما ينتج عنه المصفوفة (٢).
- لتكن المصفوفة **T** معطاة في (٣).
- $F(u,v) = \mathbf{T}.f(i,j).\mathbf{T}^T$ نتج عنه المصفوفة (٤).

(١)

$$Original = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 134 & 116 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

$F(u,v) =$

(٢)

$$M = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$

(٣)

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

(٤)

$$\begin{bmatrix} 162.3 & 40.6 & 20.0 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2.0 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6.0 & 11.5 & -6.0 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12.0 \\ -10.0 & 11.2 & 7.8 & -16.3 & 21.5 & 0.0 & 5.9 & 10.7 \end{bmatrix}$$

# مثال ٢ عن طريقة عمل الترميز JPEG

- لدينا فيما يلي مصفوفتي تكميم بمعاملي جودة مختلفين، المصفوفة (٥) بمعامل جودة ١٠، والمصفوفة (٦) بمعامل جودة ٩٠.

(٥)
$$Q_{10} = \begin{bmatrix} 80 & 60 & 50 & 80 & 120 & 200 & 255 & 255 \\ 55 & 60 & 70 & 95 & 130 & 255 & 255 & 255 \\ 70 & 65 & 80 & 120 & 200 & 255 & 255 & 255 \\ 70 & 85 & 110 & 145 & 255 & 255 & 255 & 255 \\ 90 & 110 & 185 & 255 & 255 & 255 & 255 & 255 \\ 120 & 175 & 255 & 255 & 255 & 255 & 255 & 255 \\ 245 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

(٦)
$$Q_{90} = \begin{bmatrix} 3 & 2 & 2 & 3 & 5 & 8 & 10 & 12 \\ 2 & 2 & 3 & 4 & 5 & 12 & 12 & 11 \\ 3 & 3 & 3 & 5 & 8 & 11 & 14 & 11 \\ 3 & 3 & 4 & 6 & 10 & 17 & 16 & 12 \\ 4 & 4 & 7 & 11 & 14 & 22 & 21 & 15 \\ 5 & 7 & 11 & 13 & 16 & 12 & 23 & 18 \\ 10 & 13 & 16 & 17 & 21 & 24 & 24 & 21 \\ 14 & 18 & 19 & 20 & 22 & 20 & 20 & 20 \end{bmatrix}$$

- نطبق عملية التكميم عن طريق: $\hat{F}(u,v) = round(\frac{F(u,v)}{Q(u,v)})$، فينتج لدينا المصفوفة (٧).

(٧)
$$\hat{F}(u,v) = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- من أجل عملية التكميم العكسي، فيتم من خلال ضرب كل قيمة من $\hat{F}(u,v)$ بالقيمة المقابلة لـ QM، ينتج عن هذا $\tilde{F}(u,v)$.
- للعودة الى المجال الزمني نطبق المعادلة:

$$\text{IDCT: } \tilde{f}(i,j) = \mathbf{T}^T.\tilde{F}(u,v).\mathbf{T} + \mathbf{128}$$

# مثال ٣ عن طريقة عمل الترميز JPEG

$$C_Q = \begin{bmatrix} 42 & 16 & 0 & 0 & 0 & 0 & 0 & 0 \\ -21 & -15 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 3 & -3 & 0 & 0 & 0 & 0 & 0 \\ -2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- لتكن المصفوفة التالية CQ هي المصفوفة الخاصة بالـ (Luminance) بعد تطبيق التحويل (DCT) ومن ثم عملية التكميم. على فرض أن قيمة المعامل (DC) للبلوك السابق هي ٤٠.
  - ✓ س١: ما هو ترميز قيم الـ DC لتلك المصفوفة علماً أن JPEG مستخدم؟
  - ✓ س٢: باستخدام جداول JPEG Huffman الافتراضية، قم بحساب رموز Huffman الخاصة بها (ACs)واعرض الخرج النهائي الناتج (bitstream)
  - ✓ س٣: على فرض أن قيم البكسلات الأصلية هي ٨ بت ، ما هي نسبة الضغط لهذا الترميز على فرض أن قيم المركبات (ACs) تحتاج ٨٢ بت؟

JPEG VLCs for luma AC coefficients

| RL/Size | Code Length | VLC | RL/Size | Code Length | VLC |
|---|---|---|---|---|---|
| **0/0** | **4 (EOB)** | **1010** | 4/1 | 6 | 111011 |
| 0/1 | 2 | 00 | 4/2 | 10 | 1111111000 |
| 0/2 | 2 | 01 | 4/3 | 16 | 1111111110010111 |
| 0/3 | 3 | 100 | 4/4 | 16 | 1111111110011000 |
| 0/4 | 4 | 1011 | 4/5 | 16 | 1111111110011001 |
| 0/5 | 5 | 11010 | 4/6 | 16 | 1111111110011010 |
| 0/6 | 6 | 111000 | 4/7 | 16 | 1111111110011011 |
| 1/1 | 4 | 1100 | 5/2 | 10 | 1111111001 |
| 1/2 | 6 | 111001 | 5/3 | 16 | 1111111110011111 |
| 1/3 | 7 | 1111001 | 5/4 | 16 | 1111111110100000 |
| 1/4 | 9 | 111110110 | 5/5 | 16 | 1111111110100001 |
| 3/1 | 6 | 111010 | 7/2 | 11 | 11111111001 |
| 3/2 | 9 | 111110111 | 7/3 | 16 | 1111111110101111 |
| 3/3 | 11 | 11111110111 | 7/4 | 16 | 1111111110110000 |

Table for luminance DC coefficient differences.

| Category | Code length | Codeword |
|---|---|---|
| 0 | 2 | 00 |
| 1 | 3 | 010 |
| 2 | 3 | 011 |
| 3 | 3 | 100 |
| 4 | 3 | 101 |
| 5 | 3 | 110 |
| 6 | 4 | 1110 |
| 7 | 5 | 11110 |
| 8 | 6 | 111110 |
| 9 | 7 | 1111110 |
| 10 | 8 | 11111110 |
| 11 | 9 | 111111110 |

# مثال ٣ عن طريقة عمل الترميز JPEG

(١

$e_{dc}(j) = DC(j) - DC(j-1) = 42 - 40 = 2$

$e_{dc}(j) = 2 \Rightarrow Size\ category = \lceil log_2(|Val| + 1) \rceil = \lceil log_2(|2| + 1) \rceil = 1.58 = 2$

**Size=2 & Amplitude =2, The Huffman codeword is 011, bitstream for the DC coefficient in this case is 01110**

(٢

**The zig-zag scanned vector for the matrix is:**

**[ 16 −21 10 −15 0 0 0 3 −2 0 2 −3 0 0 0 0 0 2 −1 0 ▪ ▪ ]**

**⇒ (0,16), (0,-21), (0,10), (0,-15), (3,3), (0,-2), (1,2), (0,-3), (5,2), (0,-1) EOB**

$Size\ category = \lceil log_2(|Val| + 1) \rceil$

**⇒ 0/5, 0/5, 0/4, 0/4, 3/2, 0/2, 1/2, 0/2, 5/2, 0/1, 0/0**

**⇒ 11010 10000 11010 01010 1011 1010 ……..1010**

**The output bitstream for this matrix is the concatenation of DC and AC:**

**01110 11010 10000 11010 01010 1011 1010 ……..1010**

# مثال ٣ عن طريقة عمل الترميز JPEG

(٢

**The output bitstream for this matrix is the concatenation of DC and AC:**

**01110 11010 10000 11010 01010 1011 1010 .....1010**

(٣

**The total number of bits for this block is 5 + 82 = 87 bits. In contrast, the original matrix has 64 entries, each of 8 bits. Hence it requires 512 bits, giving a compression ratio of 5,9**

جَامعة المَنارة
MANARA UNIVERSITY

# نهاية المحاضرة

# أسئلة؟؟